

## **ABSTRACT OF THE DISCLOSURE**

[1092] We propose a new class of *quickly reacquirable* mutual exclusion locks (QRLs) that are optimized for the case in which a single process repeatedly acquires and releases the lock and in which no other process attempts to acquire the same lock. This case is very common in an important class of applications that includes such systems as Java™ Virtual Machines. Typically, when the first holder of a QRL first acquires the lock, it *biases* the lock to itself, although bias may be directed in different way or at different times in some realizations. Biasing typically involves a one-time *compare-and-swap* instruction. Thereafter, this *bias holder* process can reacquire and release the lock via a highly optimized *ultra fast path*. If a second process attempts to acquire a QRL, then the lock reverts to a “default” lock. We demonstrate generalized techniques that enable use of any standard mutual exclusion lock as the default lock. Finally, we discuss sample techniques to allow reinitialization of a QRL lock so that it can be rebiased. Such techniques are particularly valuable in the case of migratory data access patterns.